

What is claimed is:

Sub 1
A1

1 ~~A method comprising:~~
2 calling a scheduling driver to start an Input/Output (I/O) request to a device for an
3 application;
4 determining if the device is busy; and
5 if the device is not busy,
6 providing an estimated processing time (EPT) for the I/O request to be
7 ~~completed for the application.~~

Sub 2

2. The method of claim 1, wherein determining if the device is busy comprises
2 determining whether a locked flag is set, if the locked flag is set the device is busy and if the
3 locked flag is not set the device is not busy.

1 3. The method of claim 1, further comprising, setting a locked flag if the device
2 is not busy.

1 4. The method of claim 1, further comprising, sleeping for the estimated
2 processing time (EPT).

Sub 3

5. The method of claim 4, further comprising, calling the scheduling driver to
2 obtain I/O operation results after sleeping for the estimated processing time and determining
3 if the I/O request has been completed.

1 6. The method of claim 5, further comprising, clearing a locked flag if the I/O
2 request has been completed.

Sub
pg 1

1 ~~7. The method of claim 5, further comprising, providing the I/O operation results~~
2 ~~from the I/O request to the application if the I/O request has been completed.~~

1 8. The method of claim 5, further comprising, sleeping for a timer tick interval if
2 the I/O request has been completed.

Sub
pg 2

1 ~~9. The method of claim 5, further comprising, calculating an estimated~~
2 ~~processing time remaining (EPTR) for the I/O request to be completed, if the I/O request has~~
3 ~~not been completed, and providing the estimated processing time remaining (EPTR) to the~~
4 ~~application.~~

Sub
pg 3

1 10. The method of claim 9, further comprising:
2 sleeping for the estimated processing time remaining (EPTR);
3 calling the scheduling driver to obtain the I/O operation results after sleeping for the
4 estimated processing time (EPTR); and
5 determining if the I/O request has been completed.

1 11. The method of claim 10, further comprising:
2 determining if the I/O request has been completed and calculating an estimated
3 processing time remaining (EPTR) for the I/O request to be completed, if the I/O request has
4 not been completed;
5 sleeping for the estimated processing time remaining (EPTR);
6 calling the scheduling driver to obtain the I/O operation results after sleeping for the
7 estimated processing time (EPTR); and
8 if the I/O request has not been completed,

10 repetitively performing the above operations until the I/O request has been completed.

Sub
184

12. ~~The method of claim 1, further comprising calculating an estimated amount of~~
time left (EATL) until the device will be available to the application if the device is busy, and
providing the estimated amount of time left (EATL) to the application.

13. The method of claim 12, further comprising:
sleeping for the estimated amount of time left (EATL);
calling the scheduling driver to start the I/O request to the device for the application
after sleeping for the estimated amount of time left (EATL); and
determining if the device is still busy.

14. The method of claim 13, further comprising:
determining if the device is still busy and calculating the estimated amount of time
left (EATL) until the device will be available, if the device is still busy;
sleeping for the estimated amount of time left (EATL);
calling the scheduling driver to start the I/O request to the device for the application,
after sleeping for the estimated amount of time left (EATL); and
if the I/O request has not been started,
repetitively performing the above operations until the I/O request has been
started.

15. A machine-readable medium having stored thereon instructions, which when
executed by a machine, causes the machine to perform operations comprising:

3 calling a scheduling driver to start an Input/Output (I/O) request to a device for an
4 application;
5 determining if the device is busy; and
6 if the device is not busy,
7 providing an estimated processing time (EPT) for the I/O request to be
8 ~~completed for the application.~~

Sub A1
16. The machine-readable medium of claim 15, wherein determining if the device
2 is busy comprises determining whether a locked flag is set, if the locked flag is set the device
3 is busy and if the locked flag is not set the device is not busy.

17. The machine-readable medium of claim 15, further comprising the operation
2 of setting a locked flag if the device is not busy.

18. The machine-readable medium of claim 15, further comprising the operation
2 of sleeping for the estimated processing time (EPT).

Sub A1
19. The machine-readable medium of claim 18, further comprising the operations
2 of calling the scheduling driver to obtain I/O operation results after sleeping for the estimated
3 processing time and determining if the I/O request has been completed.

20. The machine-readable medium of claim 19, further comprising the operation
2 of clearing a locked flag if the I/O request has been completed.

Sub A1
21. ~~The machine-readable medium of claim 19, further comprising the operation~~
2 ~~of providing the I/O operation results from the I/O request to the application if the I/O request~~
3 ~~has been completed.~~

22. The machine-readable medium of claim 19, further comprising the operation of sleeping for a timer tick interval if the I/O request has been completed.

~~23. The machine-readable medium of claim 19, further comprising the operations~~
~~of calculating an estimated processing time remaining (EPTR) for the I/O request to be~~
~~completed, if the I/O request has not been completed, and providing the estimated processing~~
~~time remaining (EPTR) to the application.~~

24. The machine-readable medium of claim 19, further comprising the operations of:

- sleeping for the estimated processing time remaining (EPTR);
- calling the scheduling driver to obtain the I/O operation results after sleeping for the estimated processing time (EPTR); and
- determining if the I/O request has been completed.

1 25. The machine-readable medium of claim 24, further comprising performing the
2 operations of:

3 determining if the I/O request has been completed and calculating an estimated
4 processing time remaining (EPT_R) for the I/O request to be completed, if the I/O request has
5 not been completed;

6 sleeping for the estimated processing time remaining (EPTR);

7 calling the scheduling driver to obtain the I/O operation results after sleeping for the
8 estimated processing time (EPTR); and

9 if the I/O request has not been completed,

10 repetitively performing the above operations until the I/O request has
11 been completed.

11
Sub A1
1 ~~26. The machine-readable medium of claim 15, further comprising the operations~~

2 of calculating an estimated amount of time left (EATL) until the device will be available to
3 the application if the device is busy, and providing the estimated amount of time left (EATL)
4 to the application.

1 27. The machine-readable medium of claim 26, further comprising the operations
2 of:

3 sleeping for the estimated amount of time left (EATL);
4 calling the scheduling driver to start the I/O request to the device for the application
5 after sleeping for the estimated amount of time left (EATL); and
6 determining if the device is still busy.

1 28. The machine-readable medium of claim 27, further comprising performing the
2 operations of:

3 determining if the device is still busy and calculating the estimated amount of time
4 left (EATL) until the device will be available, if the device is still busy;
5 sleeping for the estimated amount of time left (EATL);
6 calling the scheduling driver to start the I/O request to the device for the application,
7 after sleeping for the estimated amount of time left (EATL); and
8 if the I/O request has not been started,

9 repetitively performing the above operations until the I/O request has been
10 started.

1 29. An apparatus comprising:

2 a processor having a memory connected thereto, the memory storing an application, a
3 scheduling driver, the application calling the scheduling driver to start an Input/Output (I/O)
4 request to a device;

5 the scheduling driver,

6 determining if a device is busy; and

7 if the device is not busy,

8 providing an estimated processing time (EPT) for the I/O request to be
9 completed for the application.

1 30. The apparatus of claim 29, wherein determining if the device is busy
2 comprises determining whether a locked flag is set, if the locked flag is set the device is busy
3 and if the locked flag is not set the device is not busy.

1 31. The apparatus of claim 29, wherein the scheduling driver sets a locked flag if
2 the device is not busy.

1 32. The apparatus of claim 29, wherein the application sleeps for the estimated
2 processing time (EPT).

1 33. The apparatus of claim 32, wherein the application calls the scheduling driver
2 to obtain I/O operation results after sleeping for the estimated processing time and determines
3 if the I/O request has been completed.

1 34. The apparatus of claim 33, wherein the scheduling driver clears a locked flag
2 if the I/O request has been completed.

Sub 21

1 35. The apparatus of claim 32 wherein the scheduling driver provides the I/O
2 operation results from the I/O request to the application if the I/O request has been
3 completed.

1 36. The apparatus of claim 32 wherein the application sleeps for a timer tick
2 interval if the I/O request has been completed.

1 37. The apparatus of claim 32 wherein the scheduling driver calculates an
2 estimated processing time remaining (EPTR) for the I/O request to be completed, if the I/O
3 request has not been completed, and provides the estimated processing time remaining
4 (EPTR) to the application.

1 38. The apparatus of claim 37, wherein the application:
2 sleeps for the estimated processing time remaining (EPTR);
3 calls the scheduling driver to obtain the I/O operation results after sleeping for the
4 estimated processing time (EPTR); and
5 determines if the I/O request has been completed.

1 39. The apparatus of claim 38, wherein the application:
2 determines if the I/O request has been completed;
3 sleeps for the estimated processing time remaining (EPTR) calculated by the
4 scheduling driver;
5 calls the scheduling driver to obtain the I/O operation results after sleeping for the
6 estimated processing time (EPTR); and
7 if the I/O request has not been completed,

8 repetetively performing the above operations until the I/O request has been
9 completed.

1 40. The apparatus of claim 29, wherein the scheduling driver calculates an
2 estimated amount of time left (EATL) until the device will be available to the application if
3 the device is busy, and provides the estimated amount of time left (EATL) to the application.

1 41. The apparatus of claim 40, wherein the application:
2 sleeps for the estimated amount of time left (EATL);
3 calls the scheduling driver to start the I/O request to the device for the application
4 after sleeping for the estimated amount of time left (EATL); and
5 determines if the device is still busy.

1 42. The apparatus of claim 41, wherein the application:
2 determines if the device is still busy;
3 sleeps for the estimated amount of time left (EATL) calculated by the scheduling
4 driver;
5 calls the scheduling driver to start the I/O request to the device for the application,
6 after sleeping for the estimated amount of time left (EATL); and
7 if the I/O request has not been started,
8 repetetively performing the above operations until the I/O request has been
9 started.